Contents

Chapter 1

Introduction

1.1 GNU Radio

GNU Radio[1] is a free & open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used with readily-available low-cost external RF hardware to create software-defined radios, or without hardware in a simulation-like environment. It is widely used in hobbyist, academic and commercial environments to support both wireless communications research and real-world radio systems.

A software radio is a radio system which performs the required signal processing in software instead of using dedicated integrated circuits in hardware. The benefit is that since software can be easily replaced in the radio system, the same hardware can be used to create many kinds of radios for many different transmission standards; thus, one software radio can used for a variety of applications.

GNU Radio performs all the signal processing. You can use it to write applications to receive data out of digital streams or to push data into digital streams, which is then transmitted using hardware. GNU Radio has filters, channel codes, synchronization elements, equalizers, demodulators, vocoders, decoders, and many other elements (in the GNU Radio jargon, we call these elements blocks) which are typically found in radio systems. More importantly, it includes a method of connecting these blocks and then manages how data is passed from one block to another. Extending GNU Radio is also quite easy; if you find a specific block that is missing, you can quickly create and add it.

Since GNU Radio is software, it can only handle digital data. Usually, complex baseband samples are the input data type for receivers and the output data type for transmitters. Analog hardware is then used to shift the signal to the desired center frequency. That requirement aside, any data type can be passed from one block to another - be it bits, bytes, vectors, bursts or more complex data types.

GNU Radio applications are primarily written using the Python programming language, while the supplied, performance-critical signal processing path is implemented in C++ using processor floating point extensions, where available. GNU Radio Companion(GRC) is a Simulink-like graphical tool to design signal processing flow graphs.

GNU Radio supports several radio front-ends, either natively or through additional out-of-tree modules. We will be using Ettus Research USRP platform and RTL-SDR TV tuners.

1.2 First flow-graph

Lets make our first flow-graph. Our aim is to generate a sinusoid and then plot it on a scope sink. Open GNU Radio companion by executing gnuradio-companion on a terminal. It opens a window similar to the one given in Figure 1.1. When we open a new flow graph, a default graph as shown in Figure 1.2 comes up.

1.3 IQ Modulator Board

The IQ modulator board features Linear Technology's LTC 5598[2] High Linearity Direct Quadrature Modulator, which works in the range from 5MHz to 1600MHz. It allows direct modulation of an RF signal using differential I and Q signals. The I/Q baseband inputs consist of voltage-to-current converters

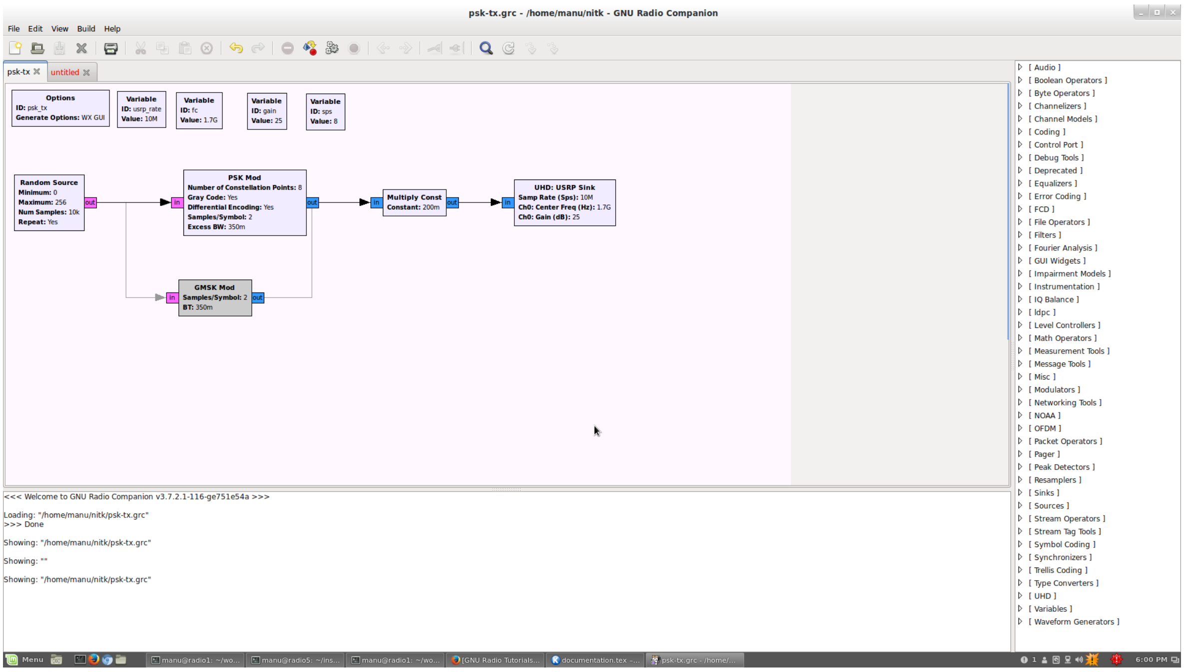

Figure 1.1: GNU Radio companion. The area containing the blocks in the center is the place where we assemble the flow-graphs. We can drag and drop different signal processing blocks to the flow-graph from the list-box on the right. The text-box on the bottom displays the output written to the terminal.

Figure 1.2: Constructing a flow graph. On the left top we see the options block. The variable block defines a variable called samp_rate. We can change the value associated with this variable by double clicking on it and editing the property window that pops up. One can use this variable in other block by putting the variable id i.e, samp_rate.

that in turn drive double-balanced mixers. The outputs of these mixers are summed and applied to a buffer, which converts the differential mixer signals to a 50Ω single-ended buffered RF output.

The carrier to the modulator is supplied from Linear Technology's LTC 6946-1[3], an ultra-low noise and spurious integer-N synthesizer with integrated VCO. This chip is a high performance, low noise phase-locked loop (PLL) with a fully integrated VCO, including a reference divider, phase-frequency detector (PFD) with phase-lock indicator, ultralow noise charge pump, integer feedback divider, and VCO output divider. It's frequency range is from 373MHz to 3740MHz. With the combination of these chips the transmitter can communicate between 373MHz to 1600MHz.

The synthesizer can be programmed through SPI interface so as to select different frequencies of operation. We use microchip's PIC18F4550 as the SPI controller and also for USB interfacing. The modulator board is designed to work either using the USB power or from external power supply.

Bibliography

[1] http://gnuradio.org

[2] http://www.linear.com/product/LTC5598

[3] http://www.linear.com/product/LTC6946